Towards a scalable, fault-tolerant, self-adaptive storage for the clouds

Houssem-Eddine Chihoub
1st year PhD student,
INRIA, Rennes - Bretagne Atlantique, France
houssem-eddine.chihoub@inria.fr

Advisor: Gabriel Antoniu
INRIA, Rennes - Bretagne Atlantique, France
gabriel.antoniu@inria.fr

Advisor: María S. Pérez-Hernández
Universidad Politécnica de Madrid, Spain
mperez@fi.upm.es

I. Introduction

The emerging model of cloud computing to manage distributed large scale infrastructure resources is becoming very popular. Entities in both industry and academia are investing a huge time and effort to investigate and develop it's features. As more and more applications are becoming data-intensive, one of the most important issues for this promising model is data management and to efficiently handle storage and I/O. In this context, data-intensive applications raise a series of challenges. For instance, designing a scalable architecture, offering a huge file sharing and fine grain access with high throughput under heavy concurrency.

This PhD is part of the FP7 European project SCALUS [1] (SCALing by means of Ubiquitous Storage). SCALUS goal is to deliver the foundation for ubiquitous storage systems. As the importance of storage is growing constantly, being able to deliver a scalable and reliable large scale distributed storage system will be critical for future IT. The purpose of this PhD within SCALUS is to design a scalable, fault tolerant, self adaptive cloud storage system.

In this PhD thesis we will address the storage problematic in IaaS (Infrastructure as a service) clouds. Cloud storage need is basically of two kinds. A storage service is essential to store and manage Virtual Machines (VMs). Such a storage support is faced with the challenge of deploying a huge number of Virtual Machine instances simultaneously in the case of several applications. Another storage service is needed for application data. For both cases, bringing high throughput under heavy concurrency is an important and challenging goal.

In a previous work [2], BlobSeer a generic data-sharing platform which aims at providing support for storing massive data with fine-grained access control under heavy concurrency on large-scale distributed infrastructures was proposed. BlobSeer stores data as huge sequences of bytes called BLOBs(Binary Large OBjects). The key features and design choices to achieve the aforementioned aim are : *Data striping*, *distributed metadata management* and *versioning based concurrency control*.

In order to enhance the Quality of Service in BlobSeer, An approach [3] was proposed. The objective was to deliver a stable throughput for individual data transfers while still achieving a high aggregated throughput in BlobSeer. To reach such an objective, a methodology based on *component monitoring*, *application-side feedback* and *behavior pattern analysis* was proposed to identify the situations that lead to fluctuations of individual data acces throughput. This approach is based on GloBeM (Global Behavior Modeling) [4], [5]. GloBeM task is to identify and describe the behavior patterns of the storage service in order to model the global behavior of large scale distributed systems.

We plan to further investigate this approach in order to propose an efficient, scalable, fault tolerant, self-adaptive storage for the clouds. This will include two main tasks that will be handled in the context of two PhD thesis. In this PhD, the focus will be on designing a storage infrastructure based on BlobSeer for the clouds and further defining a joint architecture with the global behavior modeling phase with GloBem. In the other PhD, the focus will be on designing the global behavior modeling phase to provide QoS and refining it to reach desired goals of our storage system.

II. Research plan

Storage at the application level is not quite developed in OpenNebula as there is practically no reference to it in literature. On the other hand, OpenNebula offers support for storing and managing Virtual Machines images. *Image repositories* hold the base images of the VMs and a *Virtual Machine Directory* which is a directory on the cluster node that contains running VMs. Aiming for flexibility, few configurations were proposed. One possibilty is to share the *Virtual Machine Directory* among the frontend and all cluster nodes via NFS. The other possibility would be not to share the VM Directory but to have the VMs accessible using SSH.

As these configurations show serious limitations, for instance the NFS server bottleneck when faced with concurrent deployment of VMs while the non sharing nature of SSH is a serious problem for some desired features that rely on sharing VM images like multi-versioning. One possible approach is to integrate BlobSeer as a distributed data management and storage service. As shown in [2], BlobSeer achieves a high throughput under heavy access concurrency. Each BlobSeer BLOB is split into chunks that are distributed among data providers which are essentially storage nodes. The chunks distribution follows a strategy that optimize chunk placement

in order to distribute the IO workload. In [6], a storage service based on BlobSeer to support VMs in IaaS clouds was introduced. This storage support was integrated with Nimbus compute cloud infrastructure [7]. Experiments showed encouraging improvements as storage and bandwidth usage were reduced compared to conventional approaches.

As already mentioned, the approach proposed in [3] enhanced the quality of service in BlobSeer. The approach consisted essentially in monitoring the "data providers" by collecting a wide range of parameters periodically. In order to do that GMonE monitoring framework was used [8]. The monitoring information is then gathered and aggregated inside a global history record. GloBeM automatically analyzes this history record and classify the behavior of BlobSeer into a set of states, each corresponding to a behavior pattern. After that, behavior patterns are classified into desirable and undesirable states according to feedbacks about the observed throughput and fault occurrences. Finally, predict and prevent undesired behavior patterns accordingly.

In the context of this PhD thesis, we address several storage issues: fault tolerance, scalability and self-adaptability in clouds infrastructure. We target two storage related areas, first we aim to propose an efficient storage support for storing and managing VMs in OpenNebula. In a second time, we plan to build a storage system to store, manage, and share application data in IaaS clouds.

Step 1: In order to build a storage support for VMs management, we plan to propose an approach based on the work done with BlobSeer and GloBeM [3] and optimize it to fit OpenNebula cloud requirements. Basically, we will use the approach to enhance Quality of Service in BlobSeer based on clouds requirements. Applications feedbacks and parameters to collect during the monitoring phase will be investigated. The proposed system will be integrated as a storage backend which will replace the *Virtual Machine directory*. Our approach is expected to overcome limitations already shown with existing approaches like using NFS

Moreover, our approach should offer scalable efficient support for possibly large VMs thanks to the possibility of using huge BLOBS in BlobSeer. As BlobSeer supports checkpointing also, fault tolerance will be provided via VMs checkpoints. Efficient snapshotting and multi-versioning for VMs will be supported as well. This will enable multi-versioning at the storage level rather than at the hypervisor level which should allow the hypervisor to be more efficient for other tasks. In addition, Snapshotting is less costly in BlobSeer as only the modified part of the images will be stored in a snapshotting process.

Step 2 : Our next step will be to propose an efficient, scalable, fault tolerant, self-adaptive storage system to store, manage, and share application data for IaaS clouds. VMs will be able to share data through our proposed system. Such system will be based on the same work done with BlobSeer and GloBeM and further enhanced to be adapted for clouds. Our case study for such proposal will be OpenNebula. As OpenNebula did not propose any efficient large scale storage system at the level of application data, our proposal would enhance it's general performances and it's usage. Our approach should offer a multi-versioning service to cloud clients as well.

As our system is intended to be self-adaptive thanks to GloBeM, we intend to provide storage elasticity in our proposal. One approach would be to virtualize data providers. GloBeM would be responsible then to decide wether or not to add data providers in it's analyzing phase which was already described.

Step 3: For both of our planned proposals, we intend to compare our approaches with other file and storage systems. In order to accomplish that, these systems have to be integrated in OpenNebula cloud. Systems such as HDFS, Lustre, or PVFS would be interesting references to compare with. In [9], BlobSeer performances were already compared to those of HDFS in the context of Hadoop Map-Reduce Applications. Experimentations results showed a significant improvement of the sustained throughput in scenarios that exhibit highly concurrent accesses to shared files.

We intend also to use some workloads, benchmarks and use traces in order to experiment on our approaches and evaluate their performances. Such experiments will be held on the grid 5000/ALADDIN-G5K experimental testbed [10].

Step 4: Experimenting with our approaches will provide us with results and informations on the behavior of our systems. This could be a starting point to refine the global behavior modeling phase in order to enhance our approaches and enable them to provide the desired features. Although, this step concerns more the work of the other PhD thesis.

REFERENCES

[1] SCALUS, "Scaling by means of ubiquitous storage." [Online]. Available: http://www.scalus.eu/

[2] B. Nicolae, G. Antoniu, L. Bougé, D. Moise, and A. Carpen-Amarie, "BlobSeer: Next Generation Data Management for Large Scale Infrastructures," *Journal of Parallel and Distributed Computing*, Aug. 2010.

[3] J. Montes, B. Nicolae, G. Antoniu, A. Sánchez, and M. Pérez, "Using Global Behavior Modeling to Improve QoS in Cloud Data Storage Services," in *CloudCom '10: Proc. 2nd IEEE International Conference on Cloud Computing Technology and Science*, Indianapolis, United States, Oct. 2010.

[4] J. Montes, A. Sánchez, J. J. Valdés, M. S. Pérez, and P. Herrero, "Finding order in chaos: a behavior model of the whole grid," *Concurr. Comput. : Pract. Exper.*, vol. 22, pp. 1386–1415, August 2010.

[5] ——, "The grid as a single entity towards a behavior model of the whole grid," in *Proceedings of the OTM 2008 Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008. Part I on On the Move to Meaningful Internet Systems*, ser. OTM '08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 886–897.

[6] B. Nicolae, J. Bresnahan, K. Keahey, and G. Antoniu, "Going Back and Forth: Efficient Virtual Machine Image Deployment and Snapshotting on IaaS Clouds," INRIA, Research Report RR-7482, 10 2010.

[7] "Nimbus." [Online]. Available: http://www.nimbusproject.org/

[8] A. Sánchez, "Autonomic high performance storage for grid environments based on long term perdiction," Ph.D. dissertation, Universidad Politécnica de Madrid, 2008.

[9] B. Nicolae, D. Moise, G. Antoniu, L. Bougé, and M. Dorier, "BlobSeer: Bringing High Throughput under Heavy Concurrency to Hadoop Map-Reduce Applications," in *24th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2010)*. Atlanta United States: IEEE and ACM, 02 2010.

[10] "Aladdin-g5k." [Online]. Available: http://www.grid5000.fr/

[11] OpenNebula, "The open cloud toolkit." [Online]. Available: http://opennebula.org/